

# Detector de Casas Municipales en Imágenes Aéreas (DCMIA)

Documento de sistema funcional

22 de abril de 2024



Hugo Gil Parente y Javier Martínez Madruga

[https://github.com/jamarma/AIVA\\_2024\\_DCMIA](https://github.com/jamarma/AIVA_2024_DCMIA)

# Índice

<b>Índice</b>	<b>2</b>
<b>1. Introducción</b>	<b>3</b>
1.1. Instalación . . . . .	4
<b>2. Despliegue</b>	<b>4</b>
<b>3. Funcionamiento del sistema</b>	<b>5</b>

# 1. Introducción

Se ha desarrollado un sistema de detección de casas en imágenes aéreas de alta resolución. El sistema se ejecuta en su totalidad en un servidor y se controla a través de una interfaz de usuario en un navegador web. La Figura 1 muestra un ejemplo de caso de uso.

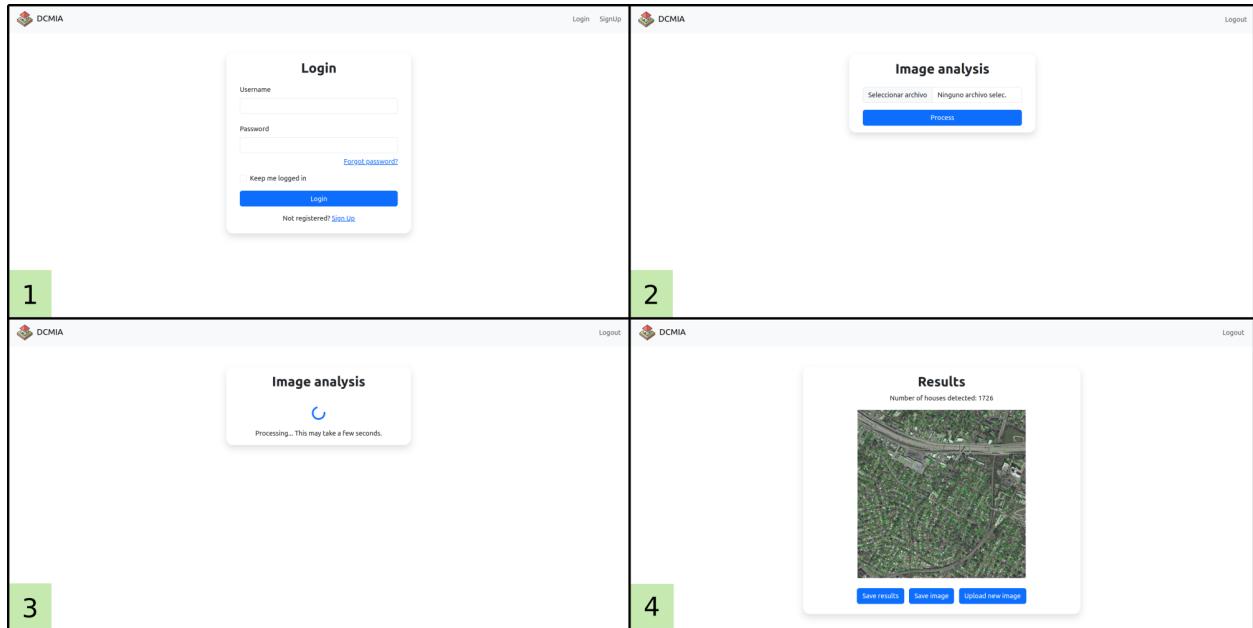
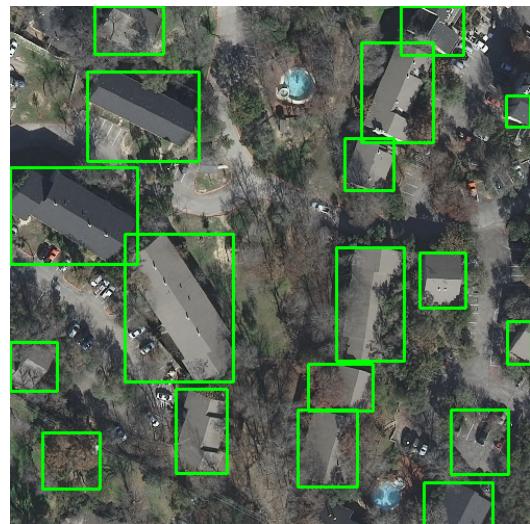


Figura 1: Ejemplo de caso de uso de la aplicación web. 1: iniciar sesión, 2: subir una imagen y solicitar su procesamiento, 3: esperar que se procese la imagen, 4: guardar los resultados.



(a) Imagen completa.



(b) Parche de la imagen completa.

Figura 2: Ejemplo de resultado ofrecido por el sistema. Se muestran los *bounding boxes* de las casas detectadas.

La aplicación web ofrece al usuario la capacidad de descargar un fichero de texto con las coordenadas de todos los *bounding boxes* detectados y el número de casas detectadas, o directamente descargar la imagen completa con los *bounding boxes* dibujados. En la Figura 2 se muestra un ejemplo visual de los resultados obtenidos con el sistema desarrollado.

## 1.1. Instalación

El proceso de instalación es sencillo. Se proporciona una imagen *Docker* con un servidor *Flask* local que corre todo el sistema. Los pasos son los siguientes:

1. **Instalar Docker.**
2. **Aceleración por GPU.** En caso de querer usar esta característica y disponer de una GPU compatible con CUDA 12.1, deberá instalar nvidia-container-toolkit.
3. **Descargar imagen *Docker*.**

```
$ docker pull jamarma/dcemia-app:latest
```

4. **Lanzar el contenedor *Docker*.** En caso de ejecutarse sin aceleración por GPU no incluir el parámetro `-gpus all`.

```
$ docker run --rm --gpus all -p 5000:5000 jamarma/dcemia-app
```

5. **Usar la aplicación.** Dirígete a `http://localhost:5000/` en tu navegador web y tendrás disponible la interfaz gráfica para usar el sistema.

## 2. Despliegue

La Figura 3 muestra el Diagrama UML de Despliegue del sistema.

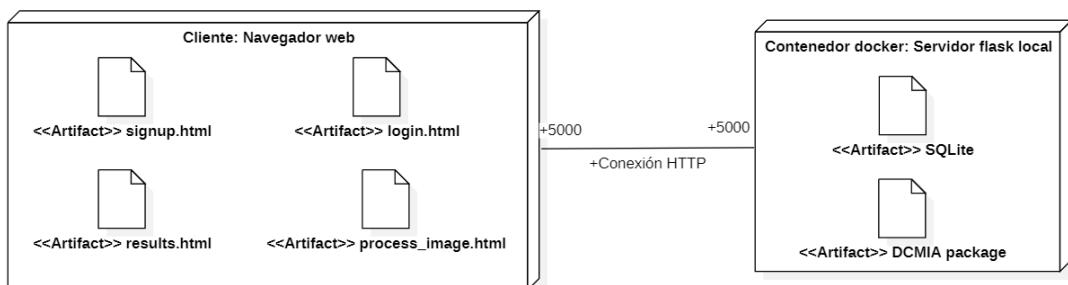


Figura 3: Diagrama UML de Despliegue.

El contenedor *Docker* actúa como servidor (en este caso local) y el navegador web actúa como cliente. En el contenedor se encuentra instalado el paquete de Python desarrollado *DCMIA*, que contiene el algoritmo de detección de casas. Además, están instaladas todas las dependencias necesarias y existe una base de datos *SQLite* para almacenar los usuarios registrados en la aplicación. Al lanzar el contenedor, se inicia un servidor web *Flask* local que se comunica con los clientes a través del puerto 5000 usando el protocolo HTTP.

### 3. Funcionamiento del sistema

La Figura 4 muestra el Diagrama UML de Secuencia del funcionamiento del sistema.

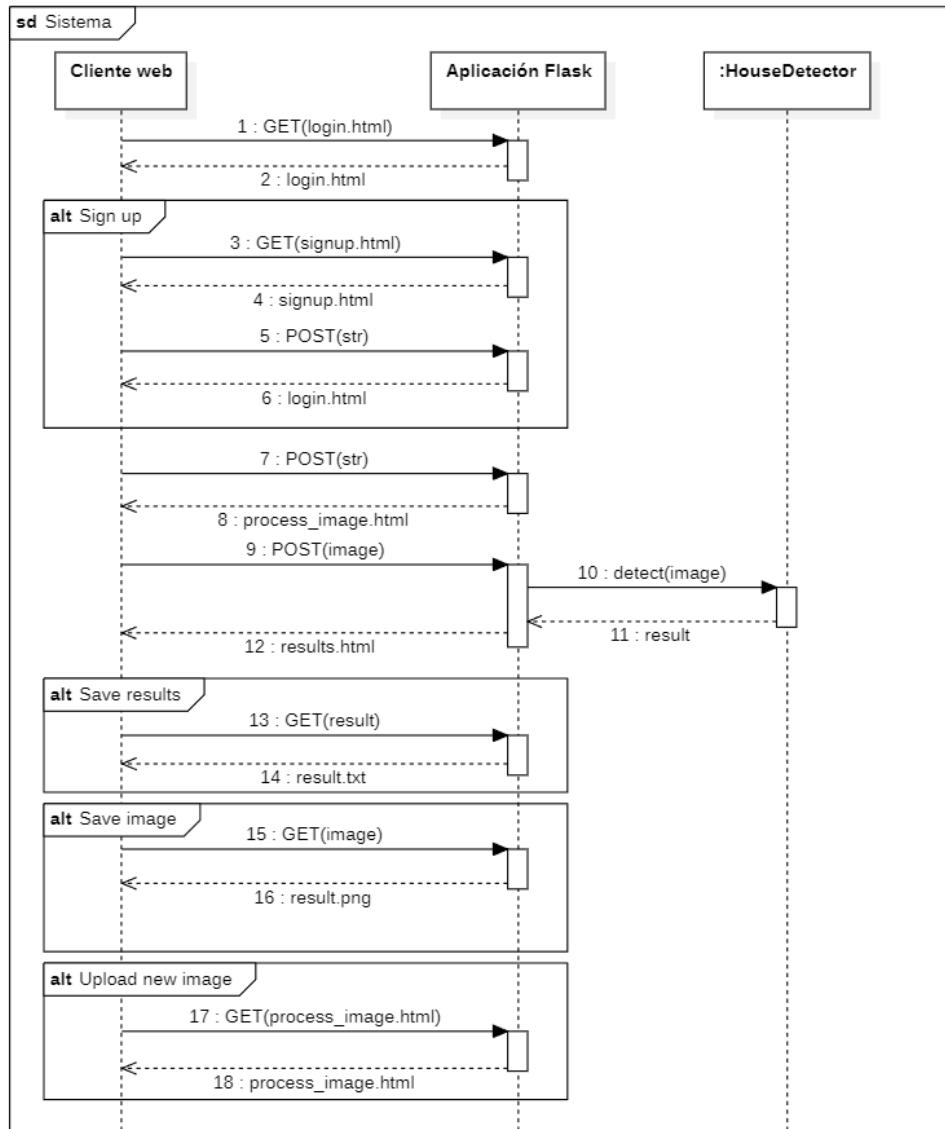


Figura 4: Diagrama UML de Secuencia del funcionamiento del sistema.

Una vez que el usuario entra en el *localhost*, el cliente web envía una solicitud al servidor y este le responde con la página de login. En ese momento el usuario deberá iniciar sesión o crearse una cuenta si todavía no la tiene. En caso de pulsar en el botón de registro, se le envía una solicitud al servidor y este responde con la página de registro. A continuación, el usuario rellena el formulario de registro y se envían los datos al servidor, recibiendo como respuesta la página de login en caso satisfactorio.

Una vez que el usuario tiene una cuenta creada, rellena el formulario de inicio de sesión, se envían los datos al servidor y, en caso de inicio de sesión exitoso, el servidor responde con la página que tiene la funcionalidad para procesar la imagen. Esta página contiene un formulario para seleccionar una imagen de los archivos locales y un botón que envía la imagen seleccionada al servidor.

Cuando el servidor recibe la imagen, usa el paquete DCMIA (objeto *HouseDetector*) para realizar la detección de casas. Una vez procesada la imagen, el servidor responde al cliente con la página de resultados que contiene el número de casas detectadas y la imagen con los *bounding boxes* dibujados.

En la página de resultados el usuario dispone de tres botones: el botón *Save results* que solicita al servidor un fichero de texto con el número de casas detectadas y las coordenadas de los *bounding boxes* detectados, el botón *Save image* que solicita al servidor la imagen con los *bounding boxes* dibujados, y el botón *Upload new image* que solicita al servidor volver a la página de procesamiento para subir una nueva imagen.